AppleScript Hjälp

Grunderna

Om AppleScript

När du använder datorn utför du många uppgifter om och om igen. Med AppleScript kan du skriva skript och program som automatiserar en del av dessa repetitiva uppgifter.

AppleScript är ett kraftfullt och mångsidigt skriptspråk som är inbyggt i Mac OS X. Med AppleScript kan du spara mycket tid genom att skapa kortkommandon, automatisera upprepade uppgifter och till och med skapa anpassade program Med AppleScript kan du styra dina skriptanpassade program.

Ett AppleScript-skript består av kommandon på ett språk som liknar engelska och som utför åtgärder på objekt. Några exempel på åtgärder är att öppna, stänga eller hämta namnet på ett objekt. Några exempel på objekt är ett program, ett fönster eller en inställning.

Ett skript kan kontrollera ett eller flera program. För att ett program skall kunna styras med skript (skriptanpassat program) så måste det ha termer definierade för sig som du kan använda i skripten. OS X levereras med flera skriptanpassade program.

Skript kan även användas för att styra program över ett nätverk. På så sätt kan du automatisera uppgifter på mer än en dator.

Faktum är att du redan har använt AppleScript utan att veta det. Många av funktionerna i systemet och i program är egentligen AppleScript-skript som anropas inifrån ett program. Ta t.ex. det här hjälpsystemet. Varje gång du klickar på en länk som öppnar ett program eller en fil är det ett AppleScript-skript som utför åtgärden.

RELATERADE ÄMNEN

Hur AppleScript fungerar
Om AppleScript-redigerare
Om AppleScript-språket
AppleScript-resurser på internet
Utvecklingsmiljön AppleScript Studio

Grunderna

Om AppleScript-redigerare

AppleScript-redigerare är ett lättanvänt skriptprogram som kan läsa, skriva, spela in och spara AppleScript-skript.

Här är några av funktionerna i AppleScript-redigerare:

- Ett enhetligt, lättanvänt gränssnitt
- Du kan hitta och ersätta text i skript
- Automatisk radbrytning av långa skriptsatser
- Du kan ångra i flera nivåer

- Kontextmenyer för snabbare skriptproduktion
- Stöd för programtjänster
- En bibliotekspalett gör det möjligt att komma åt programskript
- Skriptassistent som hjälper dig att slutföra satser
- Resultat- och händelselogghistoriken gör det möjligt att granska vad som sker i ett skript

RELATERADE ÄMNEN

Om AppleScript-språket
Utvecklingsmiljön AppleScript Studio

Grunderna

Skapa AppleScript-skript

Du kan dra nytta av AppleScript utan att kunna något om programmering alls. Du behöver inte ens skriva någon skripttext. Hundratals användbara AppleScript-skript som du kan börja använda direkt följer med OS X och tusentals fler finns tillgängliga på webben.

Om du är intresserad kan du ta befintliga skript och anpassa dem efter dina behov med några smärre ändringar.

Naturligtvis drar du störst nytta om du skapar egna skript. Om du lär dig skriva skript kan du skapa användbara verktyg och tidsbesparande automatiseringar som gör att du kan arbeta effektivare.

Du kan ändra eller skapa egna skript med AppleScript-redigerare som ger en enkel miljö för skriptskrivning. Alla exemplen i AppleScript Hjälp har gjorts med AppleScript-redigerare.

- Öppna AppleScript-redigerare.

  Skriv följande text i fönstret i AppleScript-redigerare.

  ```
  say ”Välkommen till AppleScript!”
  ```

  Klicka på Kör.

RELATERADE ÄMNEN

Om AppleScript-redigerare
Utvecklingsmiljön AppleScript Studio

Grunderna

Hur AppleScript fungerar

AppleScript fungerar genom att skicka meddelanden som kallas ”Apple-händelser”. När du kör ett skript skickar AppleScript Apple-händelser till det system eller de program som ska utföra kommandona.

Ofta returnerar program och system ett resultat till AppleScript. Om t.ex. ett skript skickar en begäran till ett program om att få namnet på ett objekt så returnerar programmet namnet. Om kommandot inte kan utföras kan programmet eller AppleScript returnera ett felmeddelande. Du kan se resultaten som returnerats i AppleScript-redigerare.

I normala fall behöver du inte känna till hur Apple-händelser agerar och fungerar. AppleScript sköter allt i bakgrunden. Det enda du behöver göra är att skriva och köra dina skript.

RELATERADE ÄMNEN

Om AppleScript
Om AppleScript-språket

Grunderna

Om appletprogram

Ett skript som sparas som ett program kallas för ett AppleScript-program eller appletprogram. Appletprogram körs automatiskt när de öppnas och utför de kommandon som finns i skriptet. Appletprogram kan vara små funktioner som bara innehåller ett par kommandon, eller avancerade program med tusentals rader.

Appletprogram kan sparas som redigerbara eller icke-redigerbara. Ett redigerbart appletprogram innehåller fortfarande skripttexten och kan därför när som helst redigeras och kompileras igen. Du redigerar och ändrar ett appletprogram genom att dra det till symbolen för AppleScript-redigerare. När du sparar ett skript som icke-redigerbart tas den redigerbara textdelen bort och filen kan inte längre redigeras eller kompileras.

Vill du redigera ett appletprogram som sparats som redigerbart öppnar du det i AppleScript-redigerare eller någon annan AppleScript-redigerare. Appletprogram kräver en ”on run”-hanterare som börjar köra skriptet när appletprogrammet öppnas.

- Detta exempel visar en ”on run”-hanterare.

```
on run
        --first command
        --second command
end run
```

RELATERADE ÄMNEN

Om dropletprogram
AppleScript-filtyper

Grunderna

Om kompilerade skript

Ett kompilerat skript är en AppleScript-fil i ett redigerbart format. Du kan läsa och redigera det i AppleScript-redigerare eller någon annan AppleScript-redigerare.

När du öppnar ett kompilerat skript visas det i AppleScript-redigerare utan att några av kommandona utförs automatiskt. Det gör att du kan göra ändringar i skriptet utan att det körs. Du kan sedan kompilera skriptet som läser och konverterar all text till AppleScript-kod i filen. Det kompilerade skriptet kan sedan köras. Ett kompilerat skript har filnamnstillägget .scpt.

När du sparar filen om ett kompilerat skript finns texten kvar i filen tillsammans med koden. Du kan därför när som helst redigera det kompilerade skriptet igen.

RELATERADE ÄMNEN

Om appletprogram
Om dropletprogram
AppleScript-filtyper

Grunderna

Om dropletprogram

Ett dropletprogram är ett AppleScript-skript som körs när du drar filer och mappar och släpper dem på programmets symbol. Skriptet utför då åtgärder på filerna och mapparna.

Precis som med appletprogram kan ett dropletprogram sparas som redigerbart eller icke-redigerbart. Ett redigerbart dropletprogram innehåller fortfarande skripttexten och kan därför när som helst redigeras och kompileras igen. Du redigerar och ändrar ett dropletprogram genom att dra det till symbolen för Skriptredigerare. När du sparar ett skript som icke-redigerbart tas den redigerbara textdelen bort, och filen kan inte längre redigeras eller kompileras.

Dropletprogram kräver en ”on open”-hanterare som börjar köra skriptet när dropletprogrammet öppnas. ”On open”-hanteraren har en parameter som innehåller en lista över de objekt som dras till dropletprogrammet. En del av dropletskriptet granskar listan, medan en annan del utför åtgärder på objekten. Under förutsättning att dropletprogrammet inte är skrivet för att utföra åtgärder på ett enda objekt, innehåller det en upprepningsloop som rullar igenom objekten i listan och utför åtgärder på dem i tur och ordning.

- Här är ett exempel på en droplet-hanterare med ”on open” och parameterlista.

```
on open (theItemsDropped)
        --first command
        --second command
end open
```

RELATERADE ÄMNEN

Om appletprogram
Om kompilerade skript
AppleScript-filtyper

Grunderna

Använda AppleScript-ordlistor

Om du vill veta vilka kommandon och objekt du kan använda till att styra ett program tittar du i det aktuella programmets ordlista. En ordlista innehåller en uppsättning termer som ett program förstår.

Du visar ordlistan för ett program i AppleScript-redigerare genom att markera programmet i fönstret Bibliotek, eller genom att dra programsymbolen till symbolen för AppleScript-redigerare.

Termerna i en ordlista är samlade i grupper med relaterade termer. Kommandon och objekt visas med fet text.

1. Välj Fönster > Bibliotek i AppleScript-redigerare.

2. Dubbelklicka på programnamnet i fönstret Bibliotek.

3. Välj en uppsättning.

   Termerna i uppsättningen visas, tillsammans med beskrivningar av dem.

AppleScript-språket är ordnat i olika kategorier:

- Uppsättningar: Grupper med termer som hör ihop. I ordlistan för AppleScript-redigerare visas uppsättningar i den första kolumnen, märkta med ett orange ”S”.
- Kommandon: begär en åtgärd eller ett resultat. Kommandon, som också kallas för ”händelser”, visas i den andra kolumnen, märkta med ett blått ”C”.
- Klasser: Skriptanpassade objekt. Klassbeskrivningar visar en lista över de egenskaper och element som klassen innehåller. Klasser visas i den andra kolumnen, märkta med ett violett ”C”.
- Egenskaper: Objekt som innehåller värden. Egenskaper visas i den tredje kolumnen, märkta med ett violett ”P”.
- Element: Objekt som finns inuti andra objekt. Exempelvis kan ett stycke innehålla många ord. Element kan visas i den tredje kolumnen, märkta med ett orange ”E”.

RELATERADE ÄMNEN

Klassdefinitioner
Kommandodefinitioner
Använda AppleScript till att styra program

ÖPPNA

AppleScript Language Guide: Commands Reference
AppleScript Language Guide: Class Reference

Lösa problem

Om du får problem med AppleScript kan du ta en titt i följande ämnen för att hitta information som löser problemet.

Om ett fel uppstår när en användare klickar på Cancel i dialogrutan Choose From List

Om ett mappkommandoskript utlöses oväntat

Om ett skript inte hittar en skiva

Klassdefinitioner

En klassdefinition beskriver egenskaperna och elementen hos ett skriptobjekt. En klass kan jämföras med ett substantiv i en mening och klassdefinitionen är som definitionen av substantivet i en ordlista.

En klassdefinition i en AppleScript-ordlista börjar normalt med termen i fetstil, följd av en kort beskrivning av termen på samma rad.

Därefter följer en lista över element eller objekt som hör till eller utgör skriptobjektet. En elementlista ingår inte alltid i klassdefinitionen, eftersom en del skriptobjekt innehåller andra skriptobjekt medan andra inte gör det.

Klassdefinitionen avslutas med en lista över egenskaper eller kvaliteter hos skriptobjektet och deras betydelse.

En egenskapspost inleds med egenskapens namn, följt av:

- En beskrivning av det dataformat som egenskapens värde lagras i, exempelvis integer (heltal), point (lista med två tal), string (textsträng) eller boolean (true eller false)
- En alternativ typindikator, [r/o], som visar att egenskapens värde är ”read-only”, eller skrivskyddad (dvs. bara kan läsas och inte ändras)
- En kort beskrivning av egenskapen.

Om ett objekt har en pluralform kan du referera till mer än ett objekt.

Om ett objekt har en lista med element innehåller det andra objekt. Elementnamnet, som du kan använda i ett skript, visas i fetstil. Texten i normal stil beskriver hur du refererar till elementet.

Objekt kan ha egenskaper. Egenskaper visas i fetstil Texten i normal stil beskriver vilken typ av värde som egenskapen innehåller, till exempel en textsträng. Egenskaper kan ha ett konstant värde eller ha ett värde från en lista. Ett skript kan få en egenskaps värde och också sätta värdet på en egenskap, såvida inte egenskapen har [r/o] (read-only) efter syntaxbeskrivningen.

RELATERADE ÄMNEN

Kommandodefinitioner
Objektdefinitioner
AppleScript-kommandon
Använda AppleScript-ordlistor

ÖPPNA

AppleScript Language Guide: Class Reference

Kommandodefinitioner

Kommandon är termer i AppleScript som utför åtgärder. Kommandon i en AppleScript-ordlista visas i mittenkolumnen, märkta med ett blått ”C”. Definitionen av ett kommando innehåller en kort beskrivning av vad kommandot gör och dess syntax. Syntaxen är det sätt på vilket du skriver kommandot i ett skript. Den kan innehålla flera delar.

Fetmarkerade ord är termer du använder i en sats. De ska skrivas exakt som de står i ordlistan.

Blå text är information som du anger. Du kan t.ex. identifiera ett objekt eller en variabel som du har definierat.

Kommandon har ibland valfria delar, vilka visas inom hakparenteser. Du måste inte använda dessa i en sats.

RELATERADE ÄMNEN

Klassdefinitioner
Använda AppleScript-ordlistor
Händelsedefinitioner
Objektdefinitioner

ÖPPNA

AppleScript Language Guide: Commands Reference

Händelsedefinitioner

En händelsedefinition beskriver ett verb eller kommando för att tillämpa en åtgärd på ett skriptobjekt. Händelser är åtgärder eller verb som objekt kan utföra eller delta i.

En händelsedefinition börjar normalt med termen, följd av en kort beskrivning av termen på samma rad.

Sedan följer en lista över de parametrar som används i utförandet av kommandot. Valfria parametrar står inom parentes, obligatoriska parametrar gör det inte.

Varje parameterpost inleds med namnet på parametern, följt av en beskrivning av dataformatet som används med parametern. Till sist följer en kort beskrivning av parametern.

Definitionen av ett kommando innehåller en kort beskrivning (den understrukna texten) av vad kommandot gör och dess syntax. Syntaxen är det sätt på vilket du skriver kommandot i ett skript. Den kan innehålla flera delar.

Fetmarkerade ord är termer du använder i en sats. De ska skrivas exakt som de står i ordlistan.

Ord i normal stil är information som du anger. Du kan t.ex. identifiera ett objekt eller en variabel som du har definierat.

Kommandon har ibland valfria delar, vilka visas inom hakparenteser. Du måste inte använda dessa i en sats.

Kommandon bearbetar objekt. Definitionen av ett objekt innehåller dess namn och en kort beskrivning (den understrukna texten) av vad objektet är. Klassen av ett objekt bestämmer vissa egenskaper hos det, till exempel vilka kommandon som kan användas med objektet.

RELATERADE ÄMNEN

Databasevents
Klassdefinitioner
Kommandodefinitioner
Objektdefinitioner
Hur AppleScript fungerar

Objektdefinitioner

Kommandon bearbetar objekt. Definitionen av ett objekt i en AppleScript-ordlista innehåller dess namn och en kort beskrivning (den understrukna texten) av vad objektet är. Klassen av ett objekt bestämmer vissa egenskaper hos det, till exempel vilka kommandon som kan användas med objektet.

Om ett objekt har en pluralform kan du referera till mer än ett objekt.

Om ett objekt har en lista med element innehåller det andra objekt. Elementnamnet, som du kan använda i ett skript, visas i fetstil. Texten i normal stil beskriver hur du refererar till elementet.

Objekt kan ha egenskaper. Egenskaper visas i blå fetstil Texten i normal, blå stil beskriver vilken typ av värde som egenskapen innehåller, till exempel en textsträng. Egenskaper kan ha ett konstant värde eller ha ett värde från en lista. Ett skript kan få en egenskaps värde och också sätta värdet på en egenskap, om inte egenskapen har ”[r/o]” (endast läsa, ”read-only”) efter syntaxbeskrivningen.

RELATERADE ÄMNEN

Använda AppleScript till att styra program
Kommandodefinitioner
Klassdefinitioner
Tala om för ett objekt vad det ska göra
Använda AppleScript-ordlistor

ÖPPNA

AppleScript Language Guide: Class Reference
AppleScript Language Guide: Egenskaper

If-kommandon

En if-sats är ett kommando som körs bara om vissa villkor uppfylls. Varje if-sats innehåller ett booleskt uttryck som ger värdet true eller false. Kommandona inom satsen körs bara om värdet på det booleska uttrycket är true. If-satsen avslutas med end if. Till exempel:

```
if amountReceived > 100 then
        set memberStatus to "special"
end if
```

Om det booleska uttrycket är true (amountReceived är större än 100) i ovanstående sats körs kommandona i satsen. Om det booleska uttrycket är false körs inte kommandona.

If-satsen kan också kombineras med en else-sats. Kombinationen kallas en if-then-else-sats. Till exempel:

```
if amountReceived > 100 then
        set memberStatus to ”special”
else
        set memberStatus to ”general”
end if
```

Om det första booleska uttrycket i satsen ovan är true (om det mottagna värdet, amountReceived, är större än 100), körs kommandona i den första delen av satsen. Om det första booleska uttrycket är false körs kommandona i den andra delen av satsen.

Du kan skriva if-satser inom if-satser om du behöver ange flera villkor.

Mer information om satser i AppleScript hittar du i AppleScript Language Guide.

RELATERADE ÄMNEN

Repeat-kommandon
Kommandodefinitioner
Skapa variabler i AppleScript
Konstanta värden

ÖPPNA

AppleScript Language Guide – If

Repeat-kommandon

Du kan använda ”repeat”-satser till att utföra åtgärder flera gånger beroende på olika förutsättningar. En ”repeat”-sats fortsätter att köra en grupp satser i en loop tills ett villkor uppfylls och loopen avslutas. Du kan skriva ”repeat”-satser inom ”repeat”-satser för att uppfylla olika villkor.

AppleScript innehåller en rad vrepeat”-satser som är lämpliga att använda för att uppfylla olika villkor.

repeat (forever)

Upprepas kontinuerligt i en loop. Använd en ”exit”-sats till att lämna loopen.

```
repeat
        -- statement
        -- statement
        if (x = 5) then exit repeat
end repeat
```

repeat (antal) times

Upprepar det angivna antalet gånger.

```
repeat 5 times
        -- statement
        -- statement
end repeat
```

repeat while

Upprepar så länge som ett angivet villkor uppfylls.

```
set x to 1
repeat while (x < 5)
        -- statement
        -- statement
        set x to (x + 1)
end repeat
```

repeat until

Upprepar tills ett angivet villkor uppfylls.

```
set x to 1
repeat until (x = 5)
        -- statement
        -- statement
        set x to (x + 1)
```

```
end repeat
```

repeat with (variable) from (startValue) to (stopValue)

Anger variabeln till startValue och ökar sedan variabeln stegvis efter varje loop. Upprepningen slutar när variabeln uppnår stopValue.

```
repeat with x from (5) to (25)
        -- statement
        -- statement
end repeat
```

repeat with (variable) in (list)

Anger variabeln till det första objektet i listan. Varje gång loopen upprepas anges variabeln till nästa objekt. Upprepningen slutar när alla objekt i listan har använts.

```
set myList to {234, 342, 681, 874}
repeat with x in myList
        -- statement
        -- statement
end repeat
```

Mer information om ”repeat”-satser i AppleScript finns i AppleScript Language Guide.

RELATERADE ÄMNEN

”If”-kommandon
Skapa variabler i AppleScript
Konstanta värden

ÖPPNA

AppleScript Language Guide – repeat

Konstanta värden

En konstant är ett reserverat ord i AppleScript som innehåller ett förutbestämt värde. AppleScript innehåller många konstanter, som du kan använda till att hämta information om objekt, göra jämförelser och aritmetiska beräkningar. Program och andra objekt kan definiera sina egna konstanter. De här konstanterna definieras i värdeklassdefinitionerna.

Vill du se värdet för en konstant klistrar du in följande exempel i AppleScript-redigerare och klickar sedan på Kör. Klicka sedan på Resultat så visas konstantens värde.

Resultatet är också en konstant. Det är en skriptkonstant som hänvisar till det värde som returneras av ett kommando.

```
get the current date
```

Här är några andra exempel på konstanter:

- Aritmetiska konstanter: pi

- Booleska konstanter: true, false
- Datum- och tidskonstanter: minutes, hours, days, weeks
- Strängkonstanter: return, space, tab
- Skriptkonstanter: anything, current application, it, me, missing value, my, result

Mer information om konstanter i AppleScript hittar du i AppleScript Language Guide.

RELATERADE ÄMNEN

Skapa variabler i AppleScript
Objektdefinitioner
Klassdefinitioner
Händelsedefinitioner

ÖPPNA

AppleScript Language Guide - constants

Grundläggande kommandon

Skapa variabler i AppleScript

En variabel är en behållare som skapats inom ett skript och används till att lagra ett värde. När du skapar variabeln anger du ett namn för den, som också kallas en identifierare. När ett skript körs kan du placera ett värde i variabeln, komma åt värdet eller ändra det. Variabler gäller bara så länge skriptet körs. När ett skript stängs går värdet förlorat.

- De här exemplen visar hur du anger ett unikt ID och tilldelar det ett värde.

```
set myVar to ”153”

set x to ”Quarterly Reports”

set x to false
```

Vill du se hur värden lagras i en variabel klistrar du in de här exemplen i AppleScript-redigerare och klickar sedan på Kör. Klicka på Resultat så visas värdet som lagras i variabeln.

Mer information om variabler i AppleScript hittar du i AppleScript Language Guide.

RELATERADE ÄMNEN

Konstanta värden
If-kommandon
Repeat-kommandon

ÖPPNA

AppleScript Language Guide – variabler

Grundläggande kommandon

Ange ett programs placering

OS X-program är ihoppackade i ”paket”. Ett paket är en särskild mapp som innehåller de mappar och filer som utgör programmet. På grund av detta kan kommandona ”path to” och ”info for” ge olika resultat för ett program.

Om du använder kommandot ”path to” för ett program slutar sökvägen oftast med ”.app”. Filnamnstillägg kanske inte visas som en del av filnamnet i Finder, men de kan användas till att identifiera en fil i ett skript.

Om du använder kommandot ”info for” till att hämta information om ett program blir egenskapen Folder i resultatet oftast ”true”. Du kan få mer information om filen och filnamnet med hjälp av andra egenskaper i resultatet, som extension hidden, name extension och displayed name.

RELATERADE ÄMNEN

path to
info for
Referera till mappar, filer och hårddiskar
choose application
Skriptkommandon i Finder

Finder-kommandon

Referera till mappar, filer och hårddiskar

Med AppleScript kan du referera till vissa mappar, filer och skivor med namn utan att ta med någon sökväg. AppleScript kan identifiera adressen för objekt som dokumentmappen (documents), skrivbordet (desktop), hemmappen (home folder) och startskivan (startup disk).

- Detta exempel visar hur du refererar till en specialmapp med dess namn i ett skript.

```
set x to the path to the desktop folder
```

RELATERADE ÄMNEN

Referera till Finder i skript
path to
info for
Skriptkommandon i Finder

Finder-kommandon

Referera till Finder i skript

När du vill använda Finder-kommandon riktar du kommandona till Finder i en ”tell”-sats på samma sätt som när du riktar kommandon till ett program.

- Här är ett exempel som använder ”tell”-satsen i ett skript. ”Tell”-satsen hänvisar funktionen ”number of windows” till ”Finder.app.”

```
tell application ”Finder.app”
```

```
        return the number of windows
    end tell
```

RELATERADE ÄMNEN

Finder
Tala om för ett objekt vad det ska göra
Referera till mappar, filer och hårddiskar
Objektdefinitioner
Skriptkommandon i Finder

ÖPPNA

AppleScript Language Guide - Object specifiers

Skriptkommandon i Finder

Här visas några kommandon som kan användas med Finder:

activate

alias file

alias list

application file

clean up

close

column

Computer-object

container

copy

count

data size

delete

desktop window

desktop

disk

document file

duplicate

eject

empty trash

exists

file

file information

find

Finder

Finder window

folder

get

icon view options

information window

Internet location file

item

list view options

make

move

moving folder window for

open

package

path to

preferences

preferences window

print

quit

Referera till mappar, filer och hårddiskar

Referera till Finder i skript

reveal

select

set

sort

trash

update

window

Finder-kommandon

Vanliga kommandon i scripttillägg

Skripttilläggen innehåller många användbara kommandon som du kan använda i dina skript.

adding folder items to

beep

choose application

choose color

choose file

choose file name

choose folder

choose from list

choose URL

clipboard info

close access

closing folder window for

current date

delay

dialog reply

display

display dialog

do shell script

download

FTP item

get eof

info for

list disks

list folder

load script

mount volume

offset

open for access

open location

opening folder

path to

POSIX path

random number

read

round

run script

say

scripting components

set eof

set the clipboard to

set volume

store script

summarize

system attribute

time to GMT

upload

URL

write